Введение в нейронные сети

Урок 7. Детектирование объектов

Сделайте краткий обзор любой научной работы, посвящённой алгоритму для object detection, не рассматривавшемуся на уроке. Проведите анализ: чем отличается выбранная вами архитектура нейронной сети от других? В чём плюсы и минусы данной архитектуры? Какие могут возникнуть трудности при применении этой архитектуры на практике?

Давайте обсудим архитектуру RetinaNet для object detection. Она представляет собой одну из важных моделей, которая решает проблему неравномерного распределения объектов различных размеров на изображении.

Обзор:

RetinaNet, предложенный в 2017 году Lin и коллегами из Facebook AI Research, стал важным вкладом в область object detection. Архитектура RetinaNet объединяет в себе две основные идеи: эффективное обнаружение объектов различных размеров и решение проблемы неравномерного распределения классов.

RetinaNet использует Feature Pyramid Network (FPN) для создания многоуровневой пирамиды признаков, которая обеспечивает масштабную инвариантность и позволяет обнаруживать объекты различных размеров. Кроме того, для устранения проблемы неравномерного распределения классов (class imbalance) в данных, RetinaNet использует новую функцию потерь, которая фокусируется на обнаружении объектов редких классов.

Архитектура свёрточной нейронной сети (CHC) RetinaNet состоит из 4 основных частей, каждая из которых имеет своё назначение:

a) Backbone – основная (базовая) сеть, служащая для извлечения признаков из поступающего на вход изображения. Данная часть сети является вариативной и в её основу могут входить классификационные нейросети, такие как ResNet, VGG, EfficientNet и другие;

b) Feature Pyramid Net (FPN) – свёрточная нейронная сеть, построенная в виде пирамиды, служащая для объединения достоинств карт признаков нижних и верхних уровней сети, первые имеют высокое разрешение, но низкую семантическую, обобщающую способность; вторые — наоборот;

c) Classification Subnet – подсеть, извлекающая из FPN информацию о классах объектов, решая задачу классификации;

d) Regression Subnet – подсеть, извлекающая из FPN информацию о координатах объектов на изображении, решая задачу регрессии.

На рис. 1 изображена архитектура RetinaNet с ResNet нейросетью в качестве backbone.

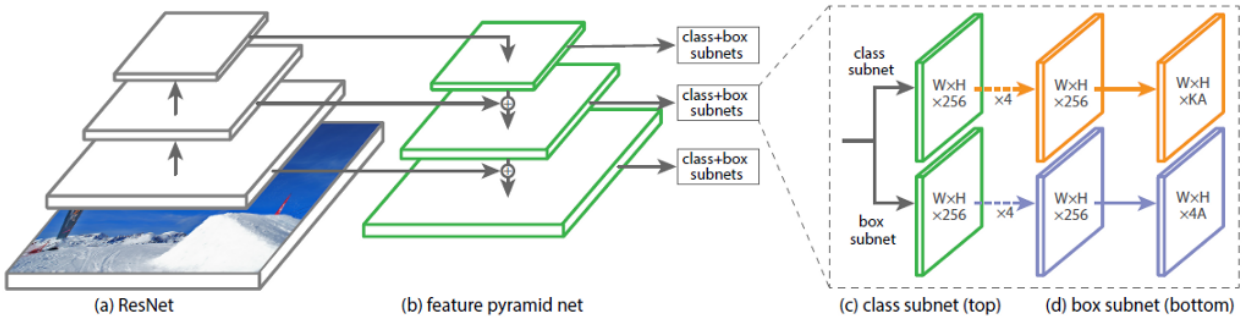

Рисунок 1 – Архитектура RetinaNet с backbone-сетью ResNet

Разберём подробно каждую из частей RetinaNet, представленных на рис. 1.

Backbone часть сети RetinaNet

Учитывая, что часть архитектуры RetinaNet, которая принимает на вход изображение и выделяет важные признаки, является вариативной и извлеченная из этой части информация будет обрабатываться на следующих этапах, то важно выбрать подходящую backbone-сеть для лучших результатов.

Недавние исследования по оптимизации СНС позволили разработать классификационные модели, которые опередили все ранее разработанные архитектуры с лучшими показателями точности на датасете ImageNet при улучшении эффективности в 10 раз. Данные сети получили название EfficientNet-B(0-7). Показатели семейства новых сетей представлены на рис. 2.

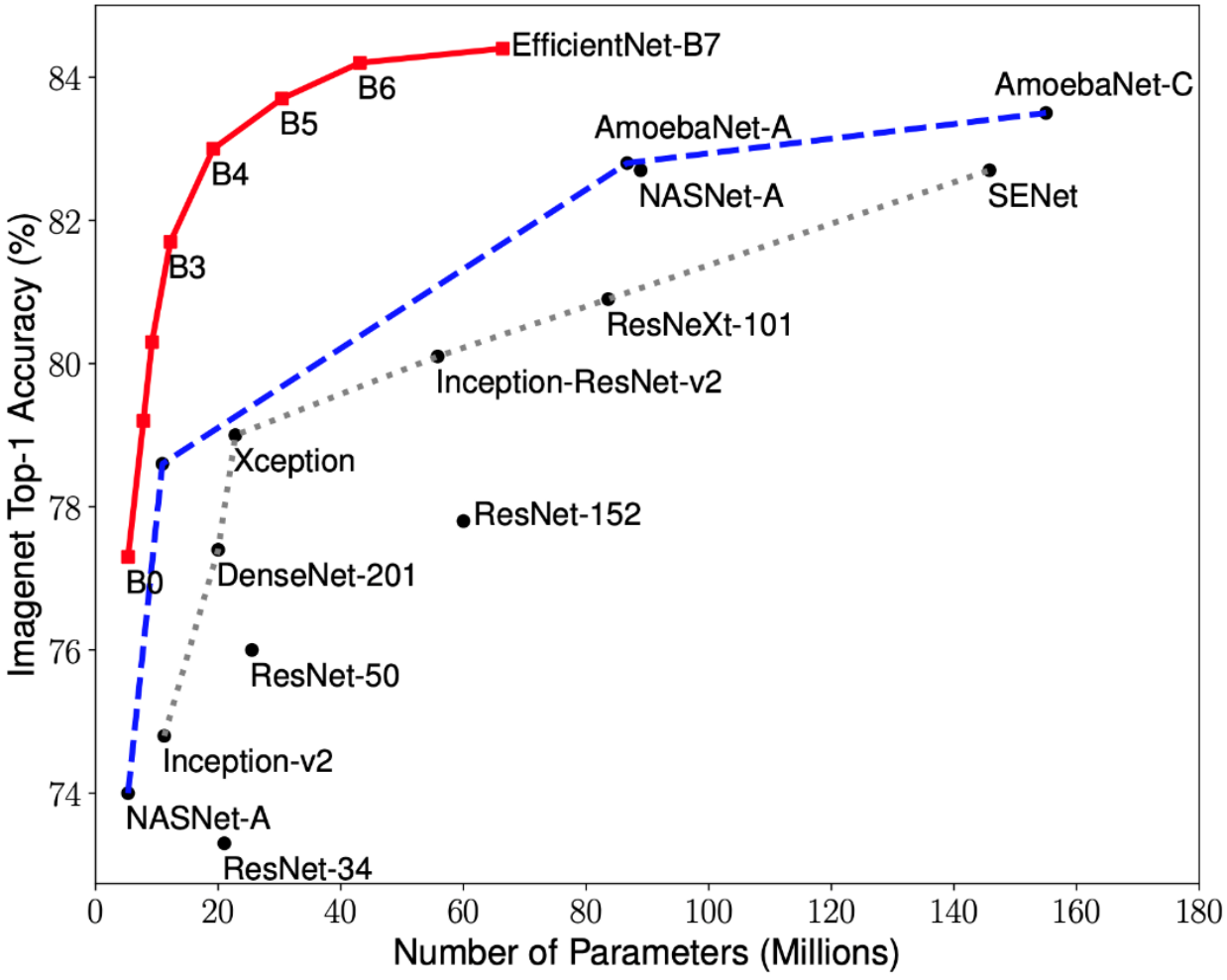

Рисунок 2 – График зависимости наибольшего показателя точности от количества весов сети для различных архитектур

Пирамида признаков

Feature Pyramid Network состоит из трёх основных частей: восходящий путь (bottom-up pathway), нисходящий путь (top-down pathway) и боковые соединения (lateral connections).
Восходящий путь представляет собой некую иерархическую «пирамиду» – последовательность свёрточных слоёв с уменьшающейся размерностью, в нашем случае – backbone сеть. Верхние слои сверточной сети имеют большее семантическое значение, но меньшее разрешение, а нижние наоборот (рис. 3). Bottom-up pathway имеет уязвимость при извлечении признаков – потеря важной информации об объекте, например из-за зашумления небольшого, но значимого, объекта фоном, так как к концу сети информация сильно сжата и обобщена.

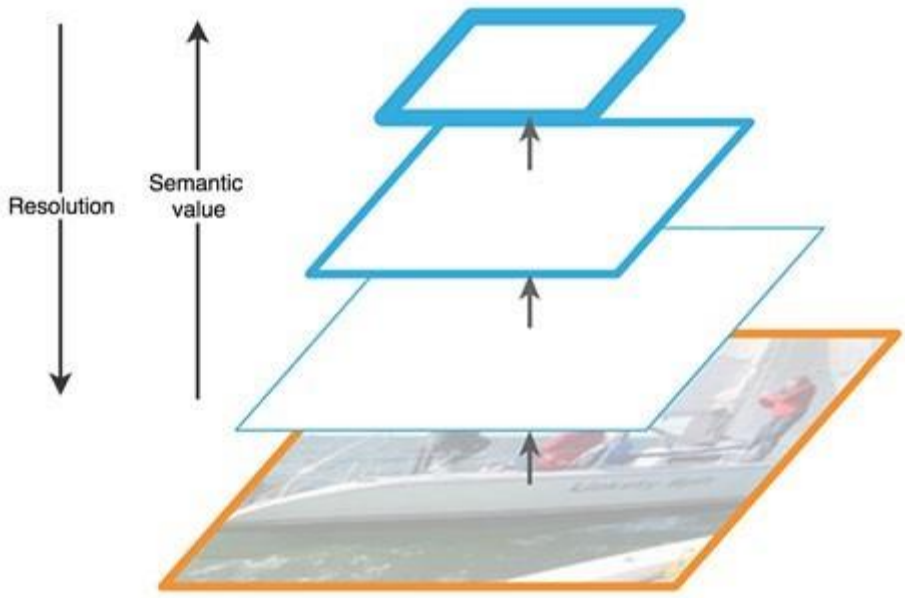

Рисунок 3 – Особенности карт признаков на разных уровнях нейросети

Нисходящий путь также представляет собой «пирамиду». Карты признаков верхнего слоя этой пирамиды имеют размер карт признаков верхнего слоя bottom-up пирамиды и увеличиваются вдвое методом ближайшего соседа (рис. 4) по направлению вниз.

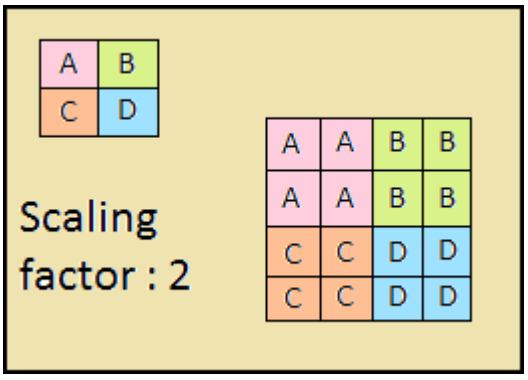

Рисунок 4 – Увеличение разрешения изображения методом ближайшего соседа

Таким образом в top-down сети каждая карта признаков вышележащего слоя увеличивается до размеров карты нижележащего. Помимо этого, в FPN присутствуют боковые соединения, это означает, что карты признаков соответствующих слоёв bottom-up и top-down пирамид поэлементно складываются, причём карты из bottom-up проходят свёртку 1*1. Этот процесс схематично представлен на рис. 5.

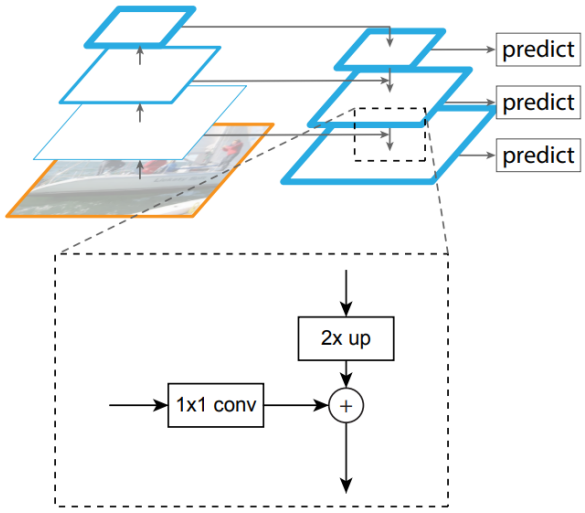

Рисунок 5 – Устройство пирамиды признаков

Боковые соединения решают проблему затухания важных сигналов в процессе прохода по слоям, совмещая семантически важную информацию, полученную к концу первой пирамиды и более детальную информацию, полученную в ней ранее.

Далее, каждый из полученных слоёв в top-down пирамиде обрабатывается двумя подсетями.

Подсети классификации и регрессии

Третей частью архитектуры RetinaNet являются две подсети: классификационная и регрессионная (рис. 6). Каждая из этих подсетей образует на выходе ответ о классе объекта и его расположении на изображении. Рассмотрим принцип работы каждой из них.

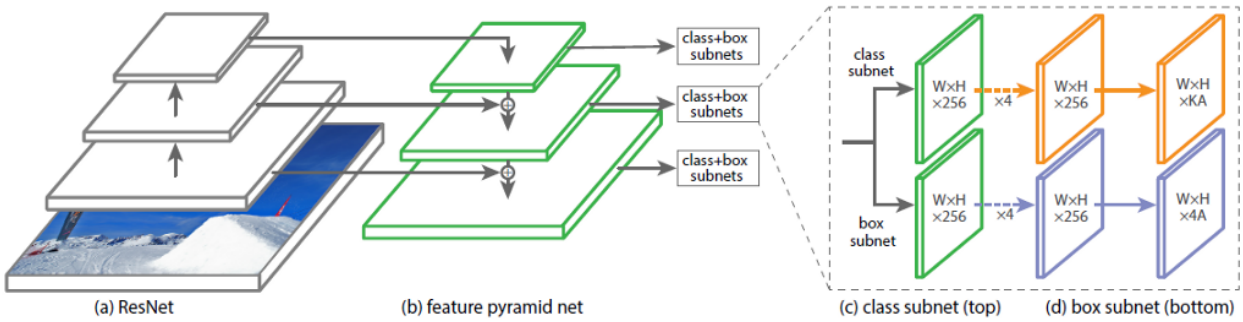

Рисунок 6 – Подсети RetinaNet

Разница в принципах работы рассматриваемых блоков (подсетей) не отличается до последнего слоя. Каждый из них состоит из 4 слоёв свёрточных сетей. В слое формируются 256 карт признаков. На пятом слое количество карт признаков изменяется: регрессионная подсеть имеет 4*A карт признаков, классификационная – K*A карт признаков, где A – количество якорных рамок (подробное описание якорных рамок в следующем подразделе), K – количество классов объектов.

В последнем, шестом, слое каждая карта признаков преобразуется в набор векторов. Регрессионная модель на выходе имеет для каждой якорной рамки вектор из 4 значений,

указывающих смещение целевой рамки (англ. ground-truth box) относительно якорной. Классификационная модель имеет на выходе для каждой якорной рамки one-hot вектор длиной K, в котором индекс со значением 1 соответствует номеру класса, который нейросеть присвоила объекту.

Якорные рамки

В прошлом разделе был использован термин якорных рамок. Якорная рамка (англ. anchor box) – гиперпараметр нейросетей-детекторов, заранее определенный ограничивающий прямоугольник, относительно которого работает сеть.

Допустим, сеть имеет на выходе карту признаков размером 3*3. В RetinaNet каждая из ячеек имеет 9 якорных рамок, каждая из которых имеет разный размер и соотношение сторон (рис. 7). Во время обучения каждой целевой рамке подбираются в соответствие якорные рамки. Если их показатель IoU имеет значение от 0.5, то якорная рамка назначается целевой, если значение меньше 0.4, то она считается фоном, в других случаях якорная рамка будет проигнорирована для обучения. Классификационная сеть обучается относительно выполненного назначения (класс объекта или фон), регрессионная сеть обучается относительно координат якорной рамки (важно отметить, что ошибка вычисляется относительно якорной, но не целевой рамки).

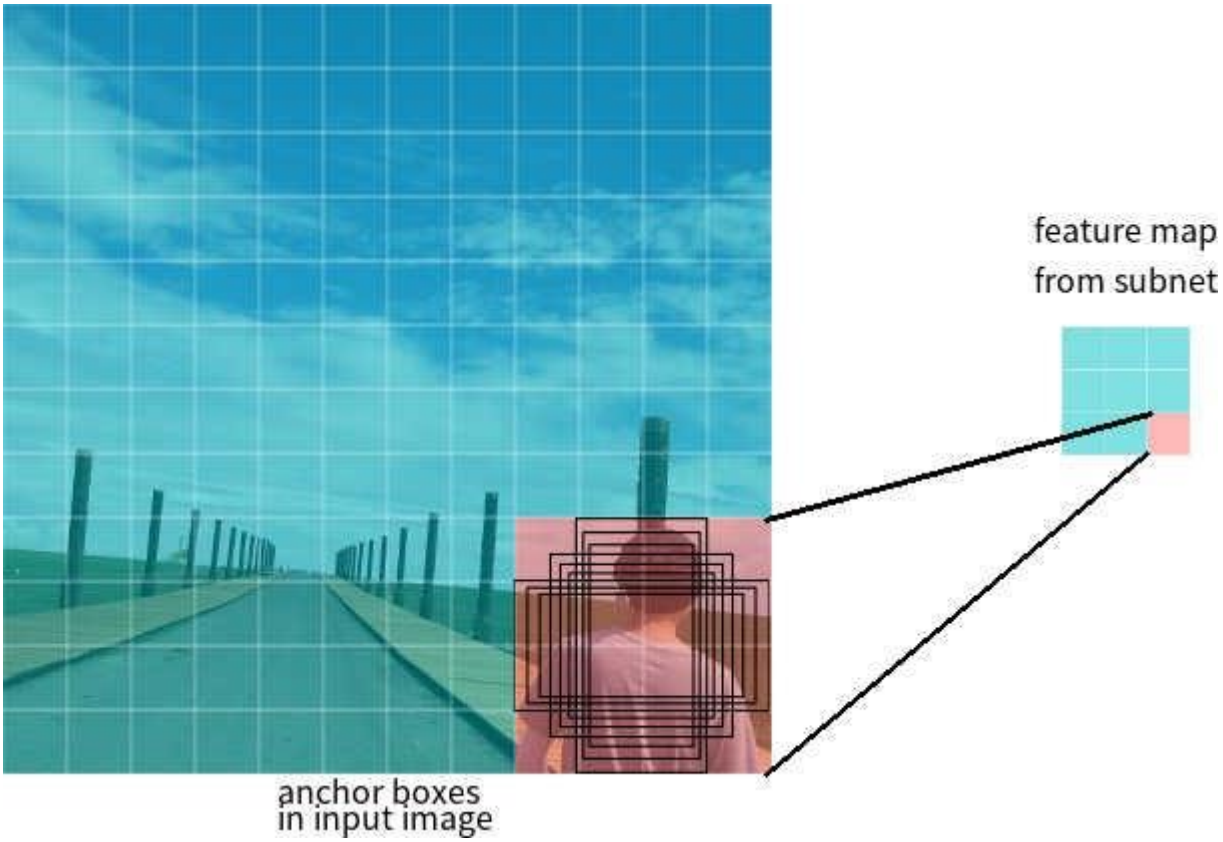

*Рисунок 7 – Якорные рамки для одной ячейки карты признаков с размером 3*3*

Функции потерь

Потери RetinaNet являются составными, их составляют два значения: ошибка регрессии, или локализации (ниже обозначено как Lloc), и ошибка классификации (ниже обозначено как Lcls). Общая функция потерь может быть записана как:

$$L = \lambda Lloc + Lcls$$

Где λ является гиперпараметром, который контролирует баланс между двумя потерями.

Рассмотрим подробнее вычисление каждой из потерь.
Как было описано ранее, каждой целевой рамке назначается якорная. Обозначим эти пары как (Ai,Gi)i=1,...N, где A представляет якорь, G – целевую рамку, а N количество сопоставленных пар.

Для каждого якоря регрессионная сеть предсказывает 4 числа, которые можно обозначить как Pi=(Pix,Piy,Piw,Pih). Первые две пары означают предсказанную разницу между координатами центров якорной Ai и целевой рамки Gi, а последние две – предсказанную разницу между их шириной и высотой. Соответственно, для каждой целевой рамки вычисляется Ti, как разница между якорной и целевой рамкой:

$$Lloc = \sum j \in \{x, y, w, h\}\, smoothL1\,(Pij - Tij)$$

Где smoothL1(x) определяется формулой ниже:

$$smooth_{\mathrm{L1}}(x) = \begin{cases} 0.5x^2, & |x| < 0 \\ |x| - 0.5, & |x| \geq 0 \end{cases}$$

Потери задачи классификации в сети RetinaNet вычисляются с помощью функции Focal loss.

$$Lcls = -i = 1Kyilog(pi)(1 - pi)\gamma$$

где K – количество классов, yi – целевое значение класса, p – вероятность предсказания i-го класса, γ – параметр фокуса, α – коэффициент смещения. Данная функция является усовершенствованной функцией кросс-энтропии. Отличие заключается в добавлении параметра γ∈(0,+∞), который решает проблему несбалансированности классов. Во время обучения, большая часть объектов, обрабатываемых классификатором, является фоном, который является отдельным классом. Поэтому может возникнуть проблема, когда нейросеть обучится определять фон лучше, чем другие объекты. Добавление нового параметра решило данную проблему, уменьшив значение ошибки для легко классифицируемых объектов. Графики функций focal и cross entropy представлены на рис.8.

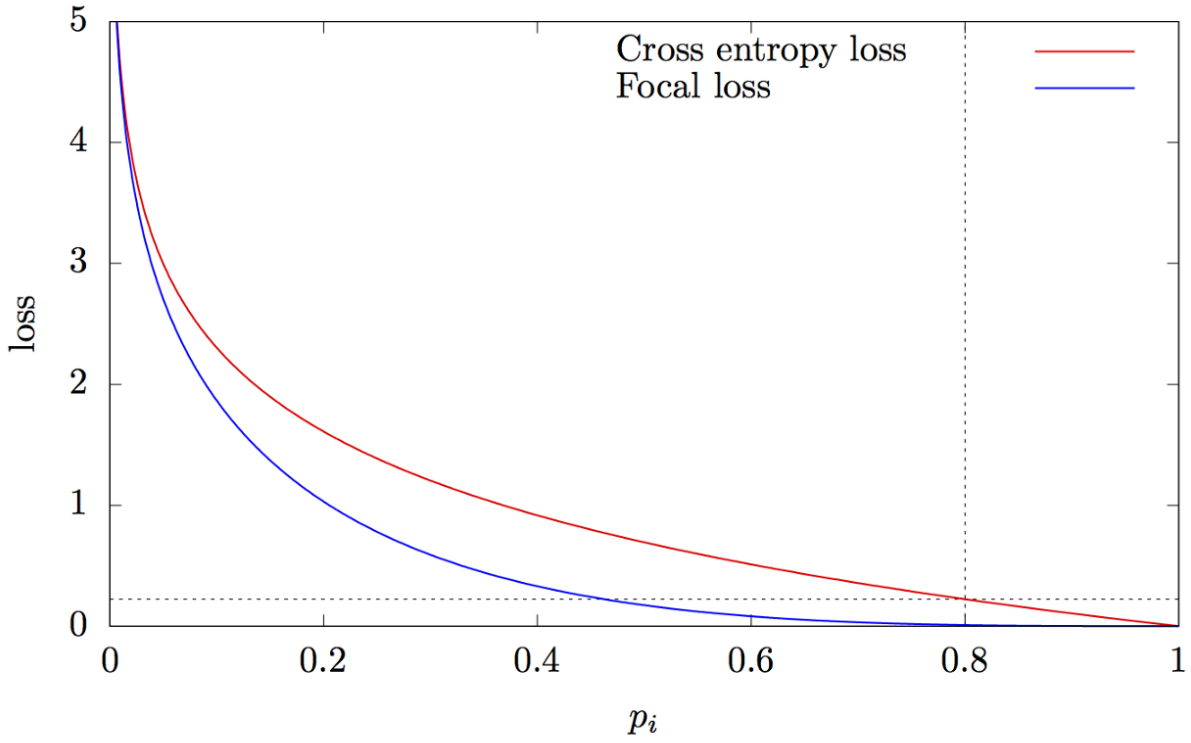

Рисунок 8 – Графики focal и cross entropy функций

Краткий анализ архитектуры:

Отличия:

Одним из ключевых отличий RetinaNet от других архитектур, таких как YOLO или SSD, является его способность эффективно обнаруживать объекты различных размеров без потери точности. Это достигается за счет использования FPN и новой функции потерь, которая обеспечивает баланс между объектами различных размеров и классами.

Плюсы и минусы:

Плюсы:

1. **Эффективное обнаружение разнообразных объектов:** RetinaNet демонстрирует высокую точность при обнаружении объектов различных размеров.
2. **Решение проблемы class imbalance:** Использование новой функции потерь позволяет справиться с проблемой неравномерного распределения классов, что улучшает общую производительность модели.
3. **Относительно простая архитектура:** По сравнению с некоторыми другими архитектурами, такими как Mask R-CNN, RetinaNet имеет более простую структуру, что может облегчить обучение и развертывание модели.

Минусы:

1. **Вычислительная сложность:** Несмотря на относительную простоту архитектуры, RetinaNet требует значительных вычислительных ресурсов для обучения и инференса из-за использования FPN и сложной функции потерь.

2. **Не учитывает контекст:** RetinaNet, как и многие другие архитектуры, не учитывает контекст объектов при их обнаружении, что может привести к ошибкам при различных сценариях.
3. **Требует больших объемов данных:** Для достижения хороших результатов RetinaNet требует больших объемов размеченных данных, что может быть проблемой для некоторых прикладных задач.

Трудности при применении:

1. **Необходимость вычислительных ресурсов:** Обучение и использование RetinaNet требует значительных вычислительных ресурсов, что может быть проблемой для малых организаций или исследователей с ограниченным бюджетом.
2. **Сложность настройки гиперпараметров:** Для достижения оптимальной производительности модели необходимо тщательно настраивать гиперпараметры, что требует времени и экспертизы.
3. **Необходимость больших объемов данных:** Для обучения RetinaNet требуется большой объем размеченных данных, что может быть сложно обеспечить в некоторых областях или для редких классов объектов.

В целом, RetinaNet представляет собой мощный инструмент для обнаружения объектов на изображениях, однако его успешное применение требует значительных усилий по обучению, настройке и обеспечению вычислительных ресурсов.